LISTING OF THE CLAIMS

Claims 1-78 are pending. No claims are amended, canceled, or withdrawn.

The following listing of claims replaces all prior versions and listings of claims in the application.

1. (Original) In a distributed computing environment, a method for dynamically implementing workflow responsive to a directory object state change, the method comprising:

detecting a state change to an object in a directory; and

responsive to detecting the state change:

mapping the state change to the object to a workflow comprising a set of tasks; and

executing the tasks to achieve a desired state in the directory.

2. (Original) A method as recited in claim 1, wherein executing the tasks further comprises storing the desired state.

3. (Original) A method as recited in claim 1, wherein executing the tasks further comprises continuously executing an operation of a task of the tasks until convergence of the desired state is identified.

4. (Original) A method as recited in claim 1, wherein executing the tasks further comprises storing a sequence of operations based on the tasks.

5. (Original) A method as recited in claim 1, wherein executing the tasks further comprises storing information corresponding to one or more directory objects to which the workflow applies.

6. (Original) A method as recited in claim 1, wherein executing the tasks further comprises storing status information based on respective status of at least one subset of the tasks.

7. (Original) A method as recited in claim 1, wherein mapping the state change to the object further comprises evaluating the state change to the object based on a declarative condition stored as a text string on an object instance of a content class defined by the directory schema.

8. (Original) A method as recited in claim 1, wherein a task of the tasks comprises any combination of a declarative condition or an operation that is stored as a text string on an object instance of a content class defined by the directory schema.

9. (Original) A method as recited in claim 1, wherein semantics of the workflow are based on a workflow schema.

10. (Original) A method as recited in claim 1, wherein mapping the state change to the object, semantics of the mapping are based on an event association object schema.

11. (Original) A method as recited in claim 1, wherein executing the tasks at least one subset of the tasks are executed with respect to one another based on an order of execution relationship comprising a finish-start relationship, a parallel execution relationship, a precedence constraint relationship, or a task priority relationship.

12. (Original) A method as recited in claim 1, wherein executing the tasks at least one subset of the tasks is executed with respect to one another based on a precedence constraint relationship or a task priority relationship.

13. (Original) A method as recited in claim 1, wherein the method further comprises:

monitoring a status corresponding to a task of the tasks;

storing the status on a status monitoring object; and

wherein a content class in the directory schema defines the status-monitoring object.

14. (Original) A method as recited in claim 1, wherein the method further comprises:

monitoring a set of directory resources affected by the workflow;

storing the directory resources on a status monitoring object; and

wherein a content class in the directory schema defines the status-monitoring object.

15. (Original) A method as recited in claim 1, wherein the method further comprises:

monitoring a status corresponding to an operation of the workflow;

determining that the status comprises a failure status;

responsive to the determining, taking a corrective action to advance the workflow in view of the failure status; and

wherein a content class in the directory schema defines the status-monitoring object.

16. (Original) A method as recited in claim 1, wherein executing the tasks further comprises:

updating a status corresponding to a task in the workflow; and

responsive to the updating, evaluating the workflow to determine that a next task of the tasks to be implemented.

17. (Original) A method as recited in claim 1, wherein the tasks represent an inverse set of tasks that were previously performed as part of a different workflow.

18. (Original) A method as recited in claim 1, wherein the tasks implement a policy with respect to one or more directory resources, and wherein mapping the state change to the object further comprises automatically determining the workflow based on the policy.

19. (Original) A computer-readable medium comprising computer-executable instructions for dynamically implementing workflow responsive to a directory object state change, the computer-executable instructions comprising instructions for:

detecting a state change to an object in a directory; and

responsive to detecting the state change:

mapping the state change to the object to a workflow comprising a set of tasks; and

executing the tasks to achieve a desired state in the directory.

20. (Original) A computer-readable medium as recited in claim 19, wherein the instructions for executing the tasks further comprise instructions for storing the desired state.

21. (Original) A computer-readable medium as recited in claim 19, wherein the instructions for executing the tasks further comprise instructions for continuously executing an operation of a task of the tasks until convergence of the desired state is identified.

22. (Original) A computer-readable medium as recited in claim 19, wherein the instructions for executing the tasks further comprise instructions for storing a sequence of operations based on the tasks.

23. (Original) A computer-readable medium as recited in claim 19, wherein instructions for executing the tasks further comprise instructions for

storing information corresponding to one or more directory objects to which the workflow applies.

24. (Original) A computer-readable medium as recited in claim 19, wherein the instructions for executing the tasks further comprise instructions for storing status information based on respective status of at least one subset of the tasks.

25. (Original) A computer-readable medium as recited in claim 19, wherein the instructions for mapping the state change to the object further comprise instructions for evaluating the state change to the object based on a declarative condition stored as a text string on an object instance of a content class defined by the directory schema.

26. (Original) A computer-readable medium as recited in claim 19, wherein a task of the tasks comprises any combination of declarative conditions and operations that are stored as a text string on an object instance of a content class defined by the directory schema.

27. (Original) A computer-readable medium as recited in claim 19, wherein semantics of the workflow are based on a workflow schema.

28. (Original) A computer-readable medium as recited in claim 19, wherein the instructions for mapping the state change to the object, semantics of the mapping are based on an event association object schema.

29. (Original) A computer-readable medium as recited in claim 19, wherein the instructions for executing the tasks, at least one subset of the tasks are executed with respect to one another based on an order of execution relationship comprising a finish-start relationship, a parallel execution relationship, a precedence constraint relationship, or a task priority relationship.

30. (Original) A computer-readable medium as recited in claim 19, wherein the instructions for executing the tasks, at least one subset of the tasks are executed with respect to one another based on a precedence constraint relationship or a task priority relationship.

31. (Original) A computer-readable medium as recited in claim 19, wherein the computer-executable instructions further comprise instructions for:

monitoring a status corresponding to a task of the tasks;

storing the status on a status monitoring object; and

wherein a content class in the directory schema defines the status-monitoring object.

32. (Original) A computer-readable medium as recited in claim 19, wherein the computer-executable instructions further comprise instructions for:

monitoring a set of directory resources affected by the workflow;

storing the directory resources on a status monitoring object; and

wherein a content class in the directory schema defines the status-monitoring object.

33. (Original) A computer-readable medium as recited in claim 19, wherein the computer-executable instructions further comprises instructions for:

monitoring a status corresponding to an operation of the workflow;

determining that the status comprises a failure status;

responsive to the determining, taking a corrective action to advance the workflow in view of the failure status; and

wherein a content class in the directory schema defines the status-monitoring object.

34. (Original) A computer-readable medium as recited in claim 19, wherein the instructions for executing the tasks further comprise instructions for:

updating a status corresponding to a task in the workflow; and

responsive to the updating, evaluating the workflow to determine that a next task of the tasks to be implemented.

35. (Original) A computer-readable medium as recited in claim 19, wherein the tasks represent an inverse set of tasks that were previously performed as part of a different workflow.

36. (Original) A computer-readable medium as recited in claim 19, wherein the tasks implement a policy with respect to one or more directory resources, and wherein the instructions for mapping the state change to the object further comprises instructions for automatically determining the workflow based on the policy.

37. (Original) A computing device comprising:

a memory comprising computer-executable instructions for dynamically implementing workflow responsive to a directory object state change; and

a processor coupled to the memory for executing the computer-executable instructions, the computer-executable instructions comprising instructions for:

detecting a state change to an object in a directory; and

responsive to detecting the state change:

mapping the state change to the object to a workflow comprising a set of tasks; and

executing the tasks to achieve a desired state in the directory.

38. (Original) A computing device as recited in claim 37, wherein the instructions for executing the tasks further comprise instructions for storing the desired state.

39. (Original) A computing device as recited in claim 37, wherein the instructions for executing the tasks further comprise instructions for continuously executing an operation of a task of the tasks until convergence of the desired state is identified.

40. (Original) A computing device as recited in claim 37, wherein the instructions for executing the tasks further comprise instructions for storing a sequence of operations based on the tasks.

41. (Original) A computing device as recited in claim 37, wherein instructions for executing the tasks further comprise instructions for storing information corresponding to one or more directory objects to which the workflow applies.

42. (Original) A computing device as recited in claim 37, wherein the instructions for executing the tasks further comprise instructions for storing status information based on respective status of at least one subset of the tasks.

43. (Original) A computing device as recited in claim 37, wherein the instructions for mapping the state change to the object further comprise instructions for evaluating the state change to the object based on a declarative condition stored as a text string on an object instance of a content class defined by the directory schema.

44. (Original) A computing device as recited in claim 37, wherein a task of the tasks comprises any combination of one or more declarative conditions and one or more operations represented by a text string stored on an object instance of a content class defined by the directory schema.

45. (Original) A computing device as recited in claim 37, wherein semantics of the workflow are based on a workflow schema.

46. (Original) A computing device as recited in claim 37, wherein the instructions for mapping the state change to the object, semantics of the mapping are based on an event association object schema.

47. (Original) A computing device as recited in claim 37, wherein the instructions for executing the tasks, at least one subset of the tasks are executed with respect to one another based on an order of execution relationship comprising a finish-start relationship, a parallel execution relationship, a precedence constraint relationship, or a task priority relationship.

48. (Original) A computing device as recited in claim 37, wherein the instructions for executing the tasks, at least one subset of the tasks are executed with respect to one another based on a precedence constraint relationship or a task priority relationship.

49. (Original) A computing device as recited in claim 37, wherein the computer-executable instructions further comprise instructions for:

monitoring a status corresponding to a task of the tasks;

storing the status on a status monitoring object; and

wherein a content class in the directory schema defines the status-monitoring object.

50. (Original) A computing device as recited in claim 37, wherein the computer-executable instructions further comprise instructions for:

monitoring a set of directory resources affected by the workflow;

storing the directory resources on a status monitoring object; and

wherein a content class in the directory schema defines the status-monitoring object.

51. (Original) A computing device as recited in claim 37, wherein the computer-executable instructions further comprises instructions for:

monitoring a status corresponding to an operation of the workflow;

determining that the status comprises a failure status;

responsive to the determining, taking a corrective action to advance the workflow in view of the failure status; and

wherein a content class in the directory schema defines the status-monitoring object.

52. (Original) A computing device as recited in claim 37, wherein the instructions for executing the tasks further comprise instructions for:

updating a status corresponding to a task in the workflow; and

responsive to the updating, evaluating the workflow to determine that a next task of the tasks to be implemented.

53. (Original) A computing device as recited in claim 37, wherein the tasks represent an inverse set of tasks that were previously performed as part of a different workflow.

54. (Original) A computing device as recited in claim 37, wherein the tasks implement a policy with respect to one or more directory resources, and

wherein the instructions for mapping the state change to the object further comprises instructions for automatically determining the workflow based on the policy.

55. (Original) A computing device comprising processing means for:

detecting a state change to an object in a directory; and

responsive to detecting the state change:

mapping the state change to the object to a workflow comprising a set of tasks; and

executing the tasks to achieve a desired state in the directory.

56. (Original) A computing device as recited in claim 55, wherein the means for executing the tasks further comprise means for storing the desired state.

57. (Original) A computing device as recited in claim 55, wherein the means for executing the tasks further comprise means for continuously executing an operation of a task of the tasks until convergence of the desired state is identified.

58. (Original) A computing device as recited in claim 55, wherein the means for executing the tasks further comprise means for storing a sequence of operations based on the tasks.

59. (Original) A computing device as recited in claim 55, wherein means for executing the tasks further comprise means for storing information corresponding to one or more directory objects to which the workflow applies.

60. (Original) A computing device as recited in claim 55, wherein the means for executing the tasks further comprise means for storing status information based on respective status of at least one subset of the tasks.

61. (Original) A computing device as recited in claim 55, wherein the means for mapping the state change to the object further comprise means for evaluating the state change to the object based on a declarative condition stored as a text string on an object instance of a content class defined by the directory schema.

62. (Original) A computing device as recited in claim 55, wherein a task of the tasks comprises any combination of one or more declarative conditions and one or more operations represented by a text string stored on an object instance of a content class defined by the directory schema.

63. (Original) A computing device as recited in claim 55, wherein semantics of the workflow are based on a workflow schema.

64. (Original) A computing device as recited in claim 55, wherein the means for mapping the state change to the object, semantics of the mapping are based on an event association object schema.

65. (Original) A computing device as recited in claim 55, wherein the means for executing the tasks, at least one subset of the tasks are executed with respect to one another based on an order of execution relationship comprising a finish-start relationship, a parallel execution relationship, a precedence constraint relationship, or a task priority relationship.

66. (Original) A computing device as recited in claim 55, wherein the means for executing the tasks, at least one subset of the tasks are executed with respect to one another based on a precedence constraint relationship or a task priority relationship.

67. (Original) A computing device as recited in claim 55, further comprising processing means for:

monitoring a status corresponding to a task of the tasks;

storing the status on a status monitoring object; and

wherein a content class in the directory schema defines the status-monitoring object.

68. (Original) A computing device as recited in claim 55, further comprising processing means for:

monitoring a set of directory resources affected by the workflow;

storing the directory resources on a status monitoring object; and

wherein a content class in the directory schema defines the status-monitoring object.

69. (Original) A computing device as recited in claim 55, further comprising processing means for:

monitoring a status corresponding to an operation of the workflow;

determining that the status comprises a failure status;

responsive to the determining, taking a corrective action to advance the workflow in view of the failure status; and

wherein a content class in the directory schema defines the status-monitoring object.

70. (Original) A computing device as recited in claim 55, wherein the means for executing the tasks further comprise means for:

updating a status corresponding to a task in the workflow; and

responsive to the updating, evaluating the workflow to determine that a next task of the tasks to be implemented.

71. (Original) A computing device as recited in claim 55, wherein the tasks represent an inverse set of tasks that were previously performed as part of a different workflow.

72. (Original) A computing device as recited in claim 55, wherein the tasks implement a policy with respect to one or more directory resources, and wherein the means for mapping the state change to the object further comprise means for automatically determining the workflow based on the policy.

73. (Previously presented) A workflow enabled directory schema comprising a plurality of base object content classes, the workflow enabled directory schema:

a provisioning service content class to detect an event corresponding to a state change in a directory object;

a workflow content class for storing a sequence of tasks;

an event association content class for storing declarative conditions to map the state change to the directory object to an object instance of the workflow content class; and

wherein the provisioning service content class is further configured to execute the sequence of tasks corresponding to the object instance.

74. (Original) A workflow enabled directory schema as recited in claim 73, wherein at least a subset of the base object content classes comprise a respective flexible attribute data field that indicates a data type, the data type being used to express various operational or data providing properties of the flexible attribute, the various operational or data providing properties being independent of the data type and independent of any modification to the workflow enabled directory schema.

75. (Original) A workflow enabled directory schema as recited in claim 73, wherein the sequence of tasks comprises any combination of a declarative conditions and operations corresponding to directory-based objects.

76. (Original) A workflow enabled directory schema as recited by claim 73, further comprising a status monitoring content class for storing a status of an object instance of the workflow content class.

77. (Original) A computer-readable medium comprising a workflow enabled directory schema as recited in claim 73.

78. (Original) A computer comprising a computer-readable medium comprising a workflow enabled directory schema as recited in claim 73.